Kinetic Spanning Trees for Minimum Power Routing in MANETS

Camillo Gentile & Robert E. Van Dyck
National Institute of Standards and Technology
Gaithersburg, Maryland
cgentile@antd.nist.gov, vandyck@antd.nist.gov

Abstract— *A distributed kinetic spanning tree algorithm is proposed for routing in wireless mobile ad hoc networks. Assuming a piecewise linear motion model for the nodes, the sequence of minimum power spanning trees is determined, while minimizing the number of routing messages required.*

Keywords— **Kinetic spanning trees, Wireless ad hoc networks, MANET**

I. INTRODUCTION

In a mobile ad hoc network (MANET), it is often necessary to route data in such a way as to minimize power consumption. For routes to a specific sink node, one can construct the minimum spanning tree [1], where the cost of each link is based on the power required. In such a tree, each node maintains in its forwarding database the next node in the tree. Because the nodes are moving, there are times at which the present spanning tree is no longer optimal, and a new minimum spanning tree should be used. This tree is typically updated using a distributed algorithm *cf.* [2], [3], [4], and it is important that the nodes be able to determine when to change their forwarding databases. To do so, messages must be exchanged among neighboring nodes. Since the energy cost of increased computation is much less than the cost of increased message transmission [5], one would like to minimize the number of routing messages exchanged.

We propose a distributed algorithm that adapts techniques from the theory of kinetic minimum spanning trees [6], [7] to maintain the correct sequence of minimum power spanning trees; additionally, the number of messages transmitted is substantially reduced, thereby providing more throughput for the data. In this paper, the power cost for transmission between two nodes varies as a square of the distance, although the proposed algorithm also works for other cost functions.

[0]This work was supported by the Advanced Research and Development Activity (ARDA) under contract number 706400.

II. PROBLEM STATEMENT

Consider the nodes in a mobile ad hoc network. Over a relatively short period of time [1], one can assume that each such node follows a linear trajectory. Its position as a function of time is described by

$$\mathbf{x}_k(t) = \begin{bmatrix} x_{k,0} + \dot{x}_k t \\ y_{k,0} + \dot{y}_k t \end{bmatrix}, \tag{1}$$

where the vector $(x_{k,0}, y_{k,0})$ gives the initial position of node k, and the vector $(\dot{x}_k, \dot{y}_k)$ gives the velocity.

The relative *squared distance* between two nodes i and j is given simply by

$$\begin{aligned} D_{ij}^2(t) &= \|\mathbf{x}_i(t) - \mathbf{x}_j(t)\|_2^2 \\ &= a_{ij}t^2 + b_{ij}t + c_{ij}, \end{aligned} \tag{2}$$

where $a \geq 0$, and $\min_t\{D_{ij}^2(t)\} = c - \frac{b^2}{4a} \geq 0$ is the minimum squared distance in time. The latter relation implies that $c \geq 0$.

Definition

The *power* as a function of time, required to transmit between nodes i and j, is defined as $P_{ij}(t) = P_{ji}(t) = \alpha D_{ij}^2(t)$, for some constant α; without loss of generality, we presently assume $\alpha = 1$. We choose power as our cost, since by minimizing this quantity through multi-hop paths, one can preserve battery life.

III. DISTRIBUTED SPANNING TREE ALGORITHM

The proposed distributed algorithm bears resemblance to the asynchronous distributed Bellman-Ford (BF) algorithm [1] for computing minimum spanning trees. With each iteration, the BF algorithm reduces the cost of the minimum multi-hop route from node i to the sink node through other nodes that comprise this route. At a single time instant, the proposed algorithm likewise reduces the costs of the minimum multi-hop routes for all time for which the fixed trajectories are valid. Whenever any node changes trajectory, it simply informs its neighbors, thereby starting a new cycle of the algorithm.

[1]The time required to transmit a data packet is orders of magnitude shorter than the time the node is moving along a fixed trajectory.

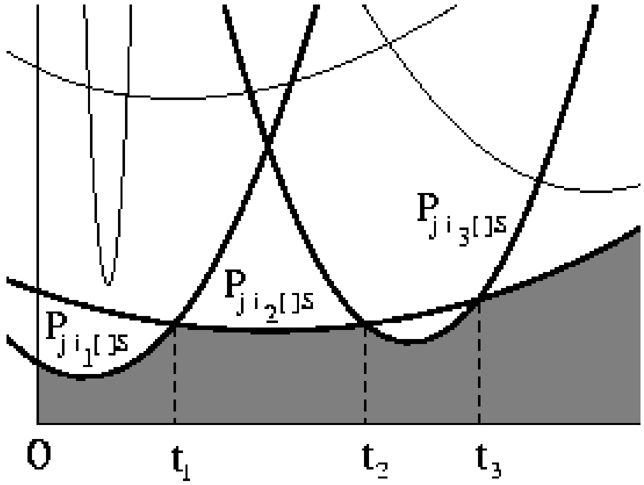

Fig. 1. New and current power costs in the pruning step at node j.

We assume that the time steps in the distributed algorithm are synchronous. The unit of time is that required to transmit a data packet from a node to any other node. Please note that the actual message exchanges are asynchronous in the sense that any node can transmit at any time (in the larger time scale). Moreover, each node need not transmit at any given time step.

Initialization

1. Each node i in the network computes $P_{iS}(t)$ and retains this *current minimum cost* in its forwarding database.
   We assume that each node can transmit to and receive from all the other nodes that are in its range as determined by the RF transmitter power of a transmitting node and the sensitivity of the receiver at the receiving node. If S is not in range, then $P_{iS}(t) = \infty$.
2. Node i computes and distributes the *new* first time cost $P_{jiS}(t) = P_{ji}(t) + P_{iS}(t)$, from node j routing through i to the sink $S, j \neq i, S$ (*i.e.* to its neighboring nodes in the network).

Iteration Step

1. At a given time step, node j receives new costs $P_{ji_1[\,]S}(t), P_{ji_2[\,]S}(t), \cdots, P_{ji_n[\,]S}(t)$, from nodes $i_1, i_2, \cdots, i_n$, which computed them at the previous iteration; $[\,]$ denotes the ordered nodes on a multi-hop route between i_l and S.
2. Pruning step: The new costs and the current costs (*i.e.* those in the forwarding database of j from the previous steps) are compared amongst each other. Only those (minimum) costs that contribute to the minimum routes of the node in time are retained in the forwarding database.
   Figure 1 visualizes both the new and current competing power costs in the pruning step at node j. The union of new and current costs appears as the six parabolic functions in time. However, only three of them, namely $P_{ji_1[\,]S}(t)$, $P_{ji_2[\,]S}(t)$, and $P_{ji_3[\,]S}(t)$, contribute to the minimum cost in time of node j. As a result, this is indicated by the shaded area. $P_{ji_1[\,]S}(t)$, $P_{ji_2[\,]S}(t)$, and $P_{ji_3[\,]S}(t)$ form the forwarding database for node j at this iteration as follows: node j forwards to node i_1 for $0 \leq t < t_1$, to node i_2 for $t_1 \leq t < t_2$, to node i_3 for $t_2 \leq t < t_3$, and to node i_2 for $t_3 \leq t$.
3. For only the new minimum costs, we compute and distribute the costs, $P_{kj[\,]S}(t) = P_{kj}(t) + P_{j[\,]S}(t)$, to node k, $k \neq j, [\,], S$. Note that the costs from previous steps were already transmitted in those steps, and need not be retransmitted.

The distributed algorithm ceases when no new minimum power costs arise at a given iteration, and so no packets need be further transmitted. At this stage, the forwarding database of each node indicates the minimum cost next-hop for times for which the fixed trajectories are valid.

We note that the number of routing packets decreases exponentially with each iteration step, both because a node can not retransmit to nodes already on its route to the sink, and because the majority of costs will not become minimum costs after the pruning step. In fact, the proposed distributed algorithm carries the same complexity and number of transmissions as the BF algorithm for minimum spanning trees; however, the proposed algorithm requires more computation at each node per iteration.

A. Efficient Distributed Algorithm

As noted above, the basic algorithm is quite similar to the distributed Bellman-Ford algorithm, except that a, b, and c of $P_{ji[]S}(t)$ are exchanged from node i to node j; in the BF algorithm, rather cost is exchanged, corresponding to the cost at a single time instance. We have also devised an efficient distributed algorithm that substantially reduces the computation at each node. The improvement is based on the theory of kinetic minimum spanning trees [6], [7] with linear costs, $W_{ij}(t)$, between two nodes, as opposed to our parabolic costs $P_{ij}(t)$.

IV. Example Network

Figure 2 shows the minimum spanning trees routed at node S for a simple five node network at two distinct times; the solid arrows indicated $t = 0$, and the dashed arrows indicate $t = 8.378$. Table I shows the trajectories for each node. We assume that these trajectories are valid

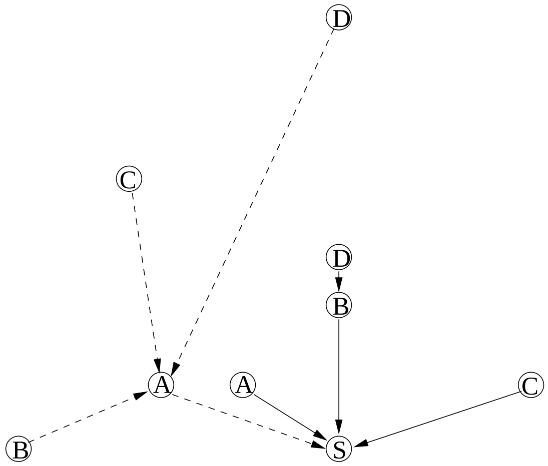

Fig. 2. Example network at $t = 0$, and $t = 8.37$.

for ten seconds. At time $t = 0$, node A transmits directly to the source. Node A is moving slowly to the left, while node B is moving more quickly down and to the left. So at time $t = 1.252$, it is more efficient for node A to route through node B, as shown in Table II. Similarly, at time $t = 2.193$, node D begins to route through node C.

Only two iterations are require for our distributed algorithm to converge, thereby providing all the nodes with the a, b, and c coefficients required to calculate the sequence of eight minimum spanning trees for the time interval $0 \leq t \leq 10$ seconds. While the distributed Bellman-Ford algorithm also converges in two iterations, it gives a minimum spanning tree for a single time instant. One can run multiple cycles of the BF algorithm in order to obtain the sequence of minimum spanning trees; yet, since the times where the minimum spanning tree changes are not known *a priori*, the BF algorithm would have to be run at a very high frequency to closely maintain the sequence. Specifically, the BF algorithm would need to be run at each time unit that corresponds to the greatest common divisor of the transition times.

Node	Trajectory
S	$(0, 0)$
A	$(-1 - 0.1t, 1)$
B	$(-0.4t, 1.5 - 0.2t)$
C	$(2 - 0.5t, 1 + 0.2t)$
D	$(0, 2 + 0.3t)$

TABLE I
NODE TRAJECTORIES.

t	Change
1.252	A → B
2.193	D → C
3.754	A → S
4.085	B → A
5.692	C → A
8.378	D → A
8.452	D → S

TABLE II
CHANGES IN THE MINIMUM SPANNING TREE.

V. COMMENTS AND CURRENT WORK

While the amount of computation is increased at each node, the number of transmissions in one cycle of the proposed algorithm is the same as in the distributed Bellman-Ford algorithm. However, the proposed algorithm need not be updated continuously, so it substantially reduces the total number of routing messages; therefore, more bandwidth is left for data messages and the total power required to send the routing messages is reduced. Moreover, by using the sequence of minimum power spanning trees for data messages, the battery life is even further increased.

In the final paper, we shall present the details of the kinetic spanning trees approach to computing the sequence of minimum spanning trees given the data exchanged in the distributed algorithm. Additionally, we shall provide extensive simulation results for much larger networks, further quantifying the performance increases achieved by the algorithm.

REFERENCES

[1] D. Bertsekas ad R. Gallager, *Data Networks*, Second Edition, Prentice-Hall, Englewood Cliffs, NJ, 1992.

[2] R. G. Gallager, P. A. Humblet, and P. M. Spira, “A distributed algorithm for minimum-weight spanning trees,” *ACM Trans. on Prog. Lang. and Systems*, Vol. 5, No. 1, pp. 66-77, Jan. 1983.

[3] A. Fetterer and S. Shekhar, “A performance analysis of hierarchical shortest path algorithms,” *Proc. IEEE Int. Conf. on Tools with Artificial Intelligence*, pp. 84-93, 1997.

[4] G. N. Frederickson, “Ambivalent data structures for dynamic 2-edge-connectivity and k smallest spanning trees,” *SIAM J. Comput.*, Vol. 26, No. 2, pp. 484-538, April 1997.

[5] G. J. Pottie, “Wireless sensor networks”, *Proc. IEEE Information Theory Workshop*, Killarney, Ireland, pp. 139-140, June 1998.

[6] P. K Agarwal, D. Eppstein, L. J. Guibas, and M. R. Henzinger, “Parametric and kinetic minimum spanning trees,” *Proc. 39th Annual Symp. on Foundations of Computer Science*, pp. 596-605

[7] L. J. Guibas, “Kinetic data structures: a state of the art report,” *Proc. 3rd Workshop Algorithmic Foundations of Robotics*, 1998.